

Ask Your Data — Business Analysis & Functional Specification

Oct 12, 2026 · @Edwin Muchalanje

Ask Your Data replaces the manual “ask an analyst” loop with self-service: a business user uploads a spreadsheet, asks a question in plain English, and gets a verified answer with a chart and the SQL behind it in about 12 seconds.

1. Purpose and scope

This document captures the business analysis behind the product: how ad-hoc data questions are handled today, the improved process, what is automated, the data model, and the functional specification the build follows.

In scope

- Uploading tabular data (CSV, Excel) and downloadable upload templates
- Natural-language questions answered by an AI analyst using read-only SQL
- Answers with a summary, chart, result table, query audit trail and follow-up suggestions
- Conversation history per dataset

Out of scope (current release)

- Joining multiple tables or connecting live databases
- User accounts, permissions and per-user rate limits
- Scheduled reports and exports

Stakeholders

Stakeholder	Interest	Role in the process
Business user (manager, sales, HR, marketing)	Fast, trustworthy answers without learning SQL	Uploads data, asks questions
Data analyst	Fewer repetitive ad-hoc requests; time for deeper work	Reviews SQL when needed; maintains templates
Product owner	Adoption, cost per question, answer accuracy	Owns requirements and KPIs

Stakeholder	Interest	Role in the process
IT / platform	Data safety, uptime, API spend	Hosts the app, manages secrets

Document control

Version	Date	Change
1.0	12 Oct 2026	First release: upload, templates, AI analyst, history

Reference: the companion *System Description* document covers architecture, components and deployment.

2. Business process mapping (as-is)

Today every ad-hoc data question goes through an analyst queue, so a one-line question waits behind other requests and often needs several rounds of clarification.

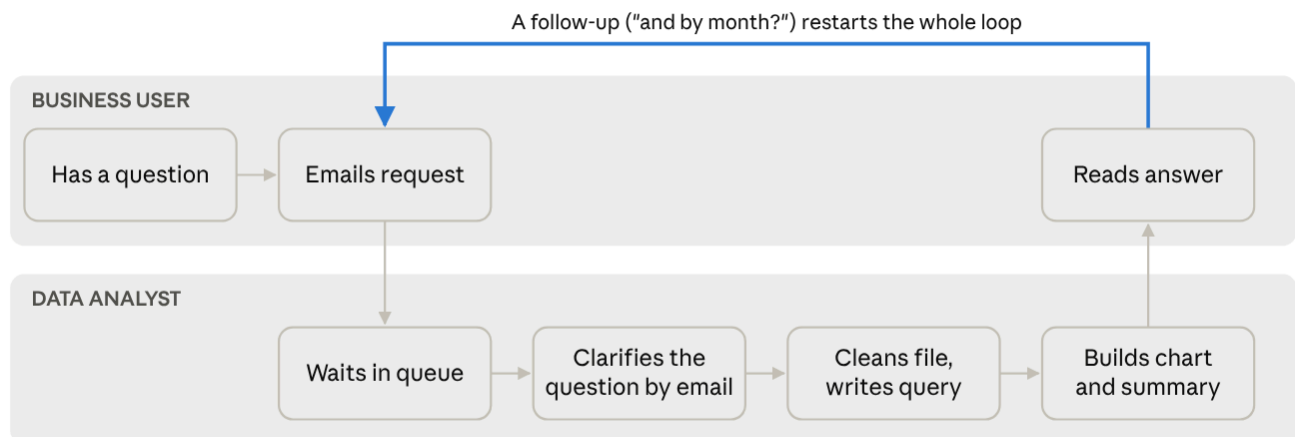
The as-is process below reflects how small and mid-size teams typically handle ad-hoc questions when data sits in spreadsheets. Durations are not quoted because they vary by team; the structural problems are the same.

#	Step	Owner	Output
1	Business user has a question about sales, staff or campaign data	Business user	Question (often vague)
2	Sends request by email or chat	Business user	Ticket / message
3	Request waits in the analyst's queue	Data analyst	Queue position
4	Analyst clarifies what is meant ("revenue before or after discount?")	Analyst + user	Agreed definition
5	Analyst finds the latest file, cleans it, writes a query or pivot	Data analyst	Working file
6	Analyst builds a chart and writes a summary	Data analyst	Email / slide
7	User reads it and asks a follow-up, restarting at step 2	Business user	New request

Pain points

Pain point	Effect	Root cause
Queue wait	Decisions delayed by the analyst's backlog	Every question needs an analyst
Clarification rounds	Several messages before work starts	Definitions agreed by email
Follow-ups restart the loop	Simple "and by month?" takes as long as the first question	No self-service path
Opaque method	Users can't see how a number was produced	Logic lives in the analyst's files
Analyst time on repetitive work	Less time for deeper analysis	Low-value requests not filtered
Version confusion	Answers from different copies of the file disagree	Data shared as email attachments

Every follow-up question goes back into the analyst queue



As-is ad-hoc request process · 2 roles, 7 steps, 1 loop

Every question crosses from the business user to the analyst and back, and a follow-up re-enters the queue.

3. Process improvement (to-be)

In the to-be process the business user answers routine questions themselves in one sitting, and the analyst only reviews work that is high-stakes or unusual.

1. Business user downloads a template (or uses an existing file) and uploads it.

2. The system profiles the data: cleans column names, detects numbers and dates, shows a preview.
3. User asks a question in plain English, or clicks a suggested question.
4. The AI analyst explores the data and computes the answer with read-only SQL.
5. User receives a summary, chart, table and the exact SQL used.
6. User asks a follow-up in the same conversation, which keeps context.
7. *Optional:* for high-stakes figures the user shares the SQL with an analyst for review.

Gap analysis

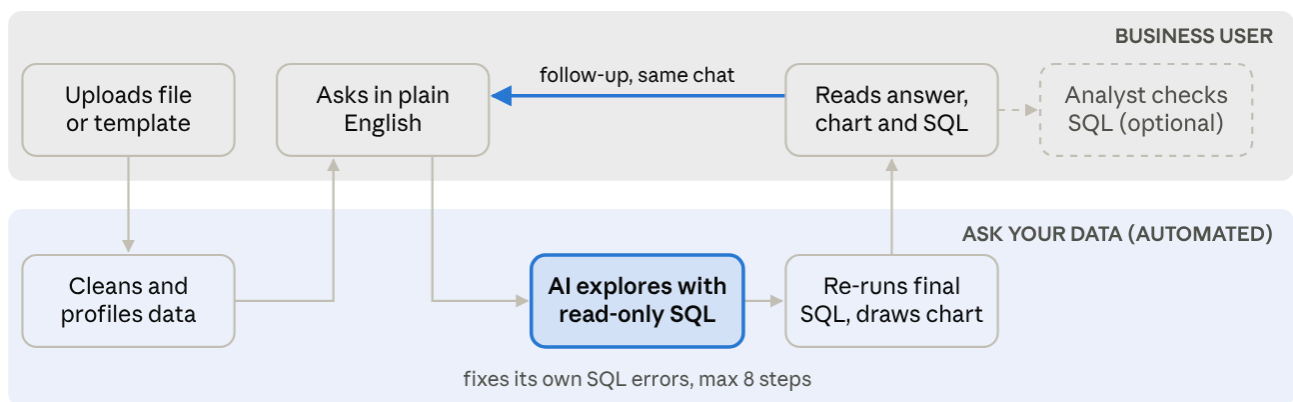
Area	As-is	To-be	How the gap is closed
Access to answers	Through an analyst	Self-service	Natural-language questions in the web app
Clarifying definitions	Email back-and-forth	AI checks distinct values, ranges and nulls first, and states its definition (e.g. "margin = profit ÷ revenue")	Exploration queries + definition in the answer
Follow-ups	Restart the request	Same conversation, recent context kept	Last 3 Q&As sent as context
Transparency	Method hidden in files	Every query shown, final SQL copyable	Audit trail per answer
Data preparation	Manual cleaning	Automatic profiling and type detection	Upload pipeline + templates
Data safety	Copies emailed around	Read-only sandbox per dataset	SQL sandbox controls (section 4)

KPIs

KPI	Target	How it is measured	Baseline
Time to answer	Under 30 s for routine questions	<code>duration_ms</code> stored per question	12 s measured on the first live question
Self-service rate	80% of routine questions answered without an analyst	Questions asked vs. analyst requests raised	To be measured after rollout

KPI	Target	How it is measured	Baseline
Answer verifiability	100% of numeric answers backed by shown SQL	Answers with non-empty <code>final_sql</code>	100% by design
Cost per question	Tracked and reported monthly	<code>input_tokens</code> + <code>output_tokens</code> per question	5,317 tokens on the first live question
Failed answers	Under 5%	Questions returning an error	To be measured

The user now gets answers and follow-ups without waiting for an analyst



To-be self-service process · user and automated system

The analyst leaves the main path; the follow-up loop now stays inside one conversation, and analyst review is an optional side step.

4. Process automation

Every step the analyst used to do by hand is automated, and the queue and hand-offs disappear; the human keeps the question and any decision taken from the answer.

As-is step	Automation	Mechanism
Find and clean the file	Automated	Upload pipeline cleans column names, converts text numbers ("1,200") and dates, stores one SQLite table per dataset
Clarify definitions	Automated	AI runs exploration queries (distinct values, date ranges, nulls) before computing

As-is step	Automation	Mechanism
Write the query	Automated	AI writes SQLite <code>SELECT</code> queries through a <code>run_sql</code> tool
Fix query errors	Automated	SQL errors are returned to the AI, which corrects and retries
Build chart and summary	Automated	AI returns a structured answer (summary, final SQL, chart type and axes, follow-ups); the app re-runs the final SQL and draws the chart
Ask the question	Human	User types or clicks a suggestion
Act on the answer	Human	User decides; optional analyst review of the SQL

Automation rules

- The AI may only call `run_sql` ; each call runs one read-only statement.
- At most 8 exploration steps per question; on the last step tools are switched off and the AI must answer with what it has.
- The answer must match a fixed JSON schema: `answer` , `final_sql` , `chart {type, x, y, title}` , `follow_ups` .
- The app re-runs `final_sql` itself; the chart and table come only from that result.
- If the chart names a column the result doesn't have, the chart is dropped and the table is shown.
- The last 3 answered questions on the dataset are passed as context for follow-ups.

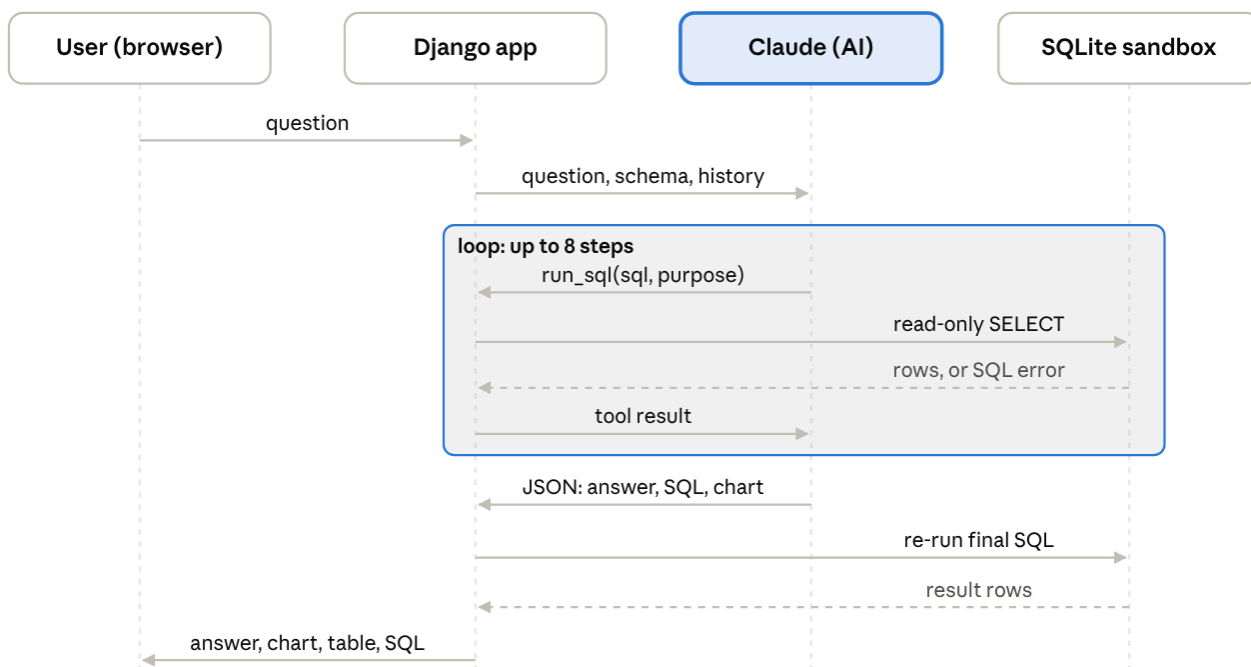
Controls

Risk	Control
AI changes or deletes data	Database opened read-only; SQLite authorizer allows only reads; <code>INSERT/UPDATE/DELETE/DROP</code> , <code>PRAGMA</code> , <code>ATTACH</code> , extensions denied
Runaway query	5-second timeout; results capped at 500 rows
Number stated without evidence	Prompt forbids it; every answer links to SQL that the app re-ran
Model declines a request	Server-side fallback model retries; otherwise a clear message is shown
API outage or bad key	Friendly error with a retry button; nothing is saved

Human checkpoints

1. The user chooses what to ask and judges whether the stated definition fits their need.
2. The "How I got this" panel lets anyone inspect every query before relying on a figure.
3. For high-stakes use (board packs, payroll), an analyst reviews the final SQL.

Claude proposes each query; the app runs it in the sandbox



Automated answer sequence · 4 participants, exploration loop of up to 8 steps

Claude never touches the data directly: it proposes SQL, the app runs it in the sandbox, and the app re-runs the final query before anything is shown.

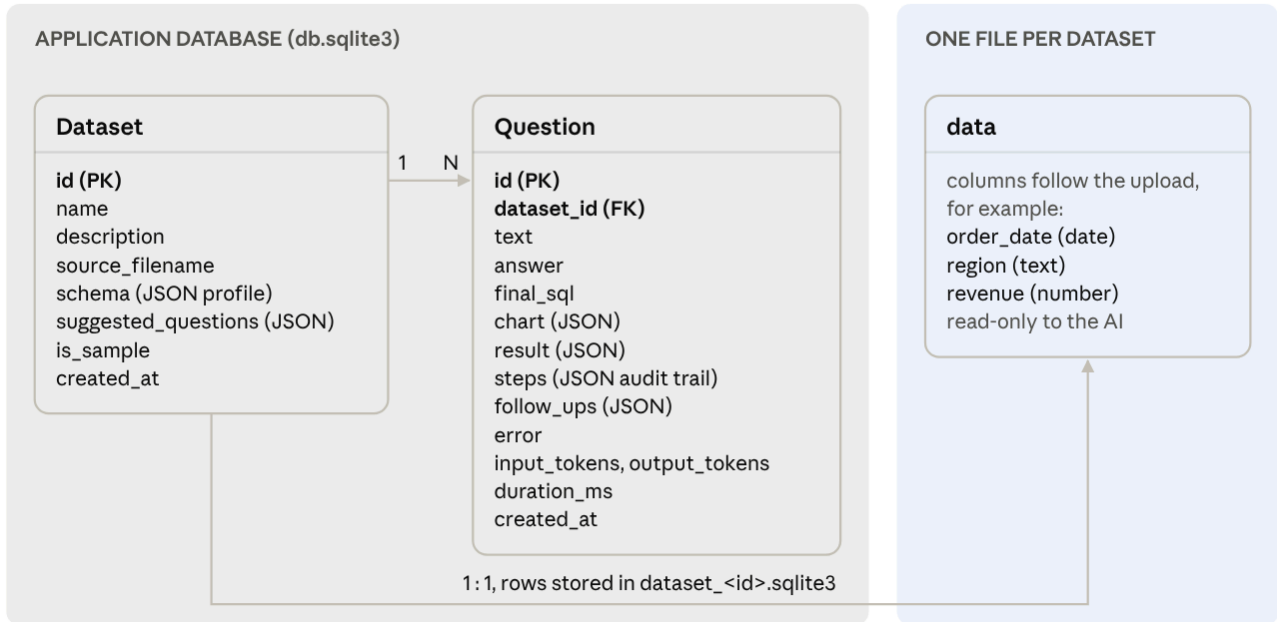
5. Data modeling

The model has two layers: an application database that records datasets and questions, and one isolated SQLite file per dataset that holds the user's data in a single table called `data`.

Conceptual model

- A **Dataset** is one uploaded file (or built-in sample). It has a profile (schema) and many Questions.
- A **Question** belongs to one Dataset and stores the answer, the SQL, the chart spec and the audit trail.
- Each Dataset owns exactly one **Data table** in its own database file.

One dataset owns many questions and exactly one data table



Entity-relationship model · 3 entities across 2 storage areas

Datasets and questions live in the application database; each dataset's rows live in their own file, which is what keeps one dataset invisible to queries about another.

Data dictionary: Dataset (application DB, table `analyst_dataset`)

Field	Type	Rules	Description
id	integer	PK, auto	Also names the data file <code>dataset_<id>.sqlite3</code>
name	text (120)	Required	Display name; defaults to the file name
description	text (300)	Optional	Shown on cards and in the AI prompt
source_filename	text (255)	Optional	Original upload name, or <code>sample</code>
schema	JSON	Required after upload	Profile: <code>row_count</code> , <code>columns[]</code> (name, type, nulls, min/max/mean or distinct + examples), <code>preview[]</code> (first 8 rows)
suggested_questions	JSON list	Optional	Starter questions shown in the sidebar

Field	Type	Rules	Description
is_sample	boolean	Default false	Samples can't be deleted
created_at	datetime	Auto	Sort order on the home page

Data dictionary: Question (application DB, table `analyst_question`)

Field	Type	Rules	Description
id	integer	PK, auto	
dataset_id	integer	FK → Dataset, cascade delete	Owning dataset
text	text	Required, ≤ 500 chars	The user's question
answer	text		Plain-English summary from the AI
final_sql	text		Query the app re-ran to build the table and chart
chart	JSON	type ∈ {bar, line, pie, none}	<code>type</code> , <code>x</code> , <code>y[]</code> , <code>title</code>
result	JSON	≤ 200 rows	<code>columns[]</code> , <code>rows[]</code> , <code>truncated</code>
steps	JSON list		Audit trail: each query's <code>sql</code> , <code>purpose</code> , <code>ok</code> , <code>row_count</code> or <code>error</code>
follow_ups	JSON list	≤ 3 items	Suggested next questions
error	text		Set when the final SQL failed on re-run
input_tokens, output_tokens	integer	≥ 0	API usage, for cost tracking
duration_ms	integer	≥ 0	End-to-end answer time
created_at	datetime	Auto	Conversation order

Physical model: per-dataset table `data`

Built from the upload at import time. The column set is dynamic and follows these rules:

Rule	Example
Names lower-cased, non-alphanumerics become <code>_</code>	<code>Order Date</code> → <code>order_date</code>
Leading digit prefixed	<code>2025 Sales</code> → <code>c_2025_sales</code>
Duplicates suffixed	<code>region</code> , <code>region_2</code>
Text that is 95%+ numeric becomes a number (commas removed)	<code>"1,200"</code> → <code>1200</code>
Text that is 90%+ date-like becomes an ISO date	<code>03/31/2025</code> → <code>2025-03-31</code>
Everything else stays text	<code>West</code>

Dates are stored as ISO text so they sort correctly and work with SQLite's `strftime()`.

6. Functional specification (FSD)

The release delivers 18 functional requirements across 7 use cases; all are built, and the core ones are covered by automated tests.

Actors

Actor	Description
Visitor / business user	Anyone with the link; uploads data and asks questions (no login in this release)
AI analyst	Claude model acting through the <code>run_sql</code> tool only
Administrator	Manages datasets and questions through the Django admin

Use cases

ID	Use case	Actor	Main flow
UC-01	Download a template	User	Home → "Download a template" → choose dataset type and Excel or CSV
UC-02	Upload a dataset	User	Choose or drag a file → optional name → Upload → dataset workspace opens

ID	Use case	Actor	Main flow
UC-03	Explore a dataset	User	View row/column counts, column types, preview rows, suggested questions
UC-04	Ask a question	User, AI analyst	Type question → AI explores with SQL → answer, chart, table, audit trail
UC-05	Ask a follow-up	User, AI analyst	Click a follow-up chip or type; prior context is used
UC-06	Verify an answer	User	Open "How I got this" → read each query → copy final SQL
UC-07	Manage datasets	User	Clear conversation; delete an uploaded dataset

Functional requirements

ID	Requirement	Priority	Use case
FR-01	The system shall offer templates for sales orders, employee records and marketing campaigns in Excel and CSV, each with 15 example rows	Should	UC-01
FR-02	Excel templates shall include an Instructions sheet describing every column and formatting rules	Should	UC-01
FR-03	The system shall accept .csv, .txt, .xlsx and .xls files up to 25 MB	Must	UC-02
FR-04	CSV import shall detect comma, semicolon and tab delimiters and UTF-8, UTF-8-BOM and Latin-1 encodings	Must	UC-02
FR-05	Import shall clean column names and detect number and date columns (rules in section 5)	Must	UC-02
FR-06	Import shall store each dataset in its own database and save a column profile	Must	UC-02
FR-07	The workspace shall show row count, column count, column names with type icons, an 8-row preview and suggested questions	Must	UC-03
FR-08	The system shall accept questions of 1-500 characters	Must	UC-04

ID	Requirement	Priority	Use case
FR-09	The AI shall answer only from query results, using up to 8 read-only SQL steps	Must	UC-04
FR-10	Each answer shall include a 2–5 sentence summary with the headline figure in bold	Must	UC-04
FR-11	The system shall re-run the final SQL and display up to 200 result rows	Must	UC-04, UC-06
FR-12	The system shall draw a line, bar or pie chart when the AI's chart spec matches the result columns, otherwise show the table only	Must	UC-04
FR-13	Each answer shall list every query run, its purpose, row count or error, and the final SQL with a copy button	Must	UC-06
FR-14	Each answer shall offer up to 3 follow-up questions as clickable chips	Should	UC-05
FR-15	Follow-ups shall include the last 3 answered questions on the dataset as context	Should	UC-05
FR-16	The system shall save every answered question and show the history when the dataset is reopened	Must	UC-04
FR-17	Users shall be able to clear a dataset's conversation and delete uploaded (non-sample) datasets	Should	UC-07
FR-18	Errors (bad file, API failure, declined request) shall show a plain-language message and a retry option	Must	UC-02, UC-04

Business rules

ID	Rule
BR-01	Generated SQL may read data but never change it
BR-02	A query may only see the dataset it was asked about
BR-03	Any number shown must come from a query the app ran
BR-04	Sample datasets cannot be deleted

ID	Rule
BR-05	A query is stopped after 5 seconds
BR-06	Pie charts are used only for parts of a whole with 6 or fewer slices

Non-functional requirements

ID	Category	Requirement
NFR-01	Performance	Typical answer in under 30 s (12 s measured)
NFR-02	Security	API key held in a private shared secrets file outside source control; never in code
NFR-03	Usability	Works on phones (390 px wide) and desktops; light and dark themes
NFR-04	Reliability	API calls retried automatically on rate limits and server errors
NFR-05	Maintainability	Model, effort level and step limit configurable by environment variable
NFR-06	Testability	Core flows tested offline with a simulated AI client

Acceptance criteria

ID	Given / When / Then	Verified by
AC-01	Given a downloaded template, when it is uploaded unchanged, then a dataset with 15 rows is created	Automated test
AC-02	Given a CSV with "1,200" in a column, when uploaded, then the column is typed as number and sums correctly	Automated test
AC-03	Given any query containing DELETE , DROP , PRAGMA or a write inside a CTE, when run, then it is rejected and data is unchanged	Automated test
AC-04	Given the AI's first query fails, when it retries, then the answer is still produced and the failed step is shown	Automated test

ID	Given / When / Then	Verified by
AC-05	Given a chart spec naming a missing column, when the answer is shown, then the table appears and no chart	Automated test
AC-06	Given a declined request, when the AI responds, then the user sees a friendly error, not a crash	Automated test
AC-07	Given "Which category has the best profit margin?" on Retail Sales, when asked, then the answer names Apparel at 53.26% with a bar chart	Manual run on live app

7. Traceability, assumptions and risks

Every pain point in section 2 traces to at least one requirement, and every Must requirement is built.

Traceability matrix

Pain point	Requirements	Process change	Status
Queue wait	FR-08, FR-09, FR-10	Self-service questions	Built
Clarification rounds	FR-07, FR-09, FR-10	AI explores first and states its definition	Built
Follow-ups restart the loop	FR-14, FR-15, FR-16	Conversation with context	Built
Opaque method	FR-11, FR-13	Audit trail and copyable SQL	Built
Analyst time on repetitive work	FR-09, FR-13	Analyst reviews SQL only when needed	Built
Version confusion	FR-01, FR-02, FR-06	Templates and one stored copy per dataset	Built

Assumptions

- Users' data fits in one table (one row per record, one header row).
- Questions are about the uploaded data, not general knowledge.
- An Anthropic API key with sufficient credit is available.

Constraints

- No user accounts in this release: anyone with the link can see all datasets.
- Upload limit 25 MB; result display limit 200 rows.

Risks

Risk	Likelihood	Impact	Mitigation
AI misreads the question and answers a different one	Medium	Medium	Definition stated in the answer; SQL shown; follow-up to correct
Public demo runs up API costs	Medium	Medium	Monthly spend limit in the Anthropic Console; per-visitor limit planned
Sensitive data uploaded to a public demo	Low	High	Planned: a warning on the upload page, and user accounts before real business use
Messy files (merged cells, totals rows) import badly	Medium	Low	Templates with instructions; preview shown after upload

Future enhancements

- Per-visitor rate limiting and user accounts with private datasets
- Live progress streaming while the AI works
- Multi-table uploads with joins; direct database connections
- Export answers as CSV or PNG; scheduled reports